

S P E C I F I C A T I O N

Docket No. **SILA0009**

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT I, **Patrick Evan Maupin**, a citizen of the United States of America, residing in the State of Texas, have invented new and useful improvements in a

METHOD FOR HANDLING DATA TRANSMISSIONS WITHIN A MODEM

of which the following is a specification:

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates to data communications in general, and in particular to data communications utilizing modems. Still more particularly, the present invention relates to a method for handling data transmissions within a modem.

2. Description of Related Art

The use of an analog modulator-demodulator (modem) to transmit digital signals across an analog channel, such as a telephone line, is well known in the art. Typically, an analog modem transmits data from a local asynchronous data terminal equipment (DTE) to a remote asynchronous DTE at the data rate of the modem's analog channel. In addition, a compression scheme is commonly utilized within a modem for storing and transmitting data in a reduced data format such that data can be communicated more efficiently. In order to compress data, a selected portion of the data must be accumulated initially. Then, the data are compressed to a selected codeword combination. When the compressed data is received at a receiving modem, the selected codeword combination is subsequently decompressed to yield the original data.

A compression scheme promulgated by the International Telecommunications Union (ITU) for data communications over a telephone network is known as the V.42*bis* standard, the pertinent of which is incorporated herein by reference. The compression scheme under the V.42*bis* standard includes two transmission modes: a transparent mode and a compressed mode. There are commands that can be sent to a decoder in both transmission modes for mode switching. For example, when in the transparent mode, a two-character sequence, which begins with a special escape character followed by an ECM (*i.e.*, enter compression mode) command, instructs the decoder to switch to the compressed

1 mode. When in the compressed mode, an ETM (*i.e.*, enter transparent mode) command
2 instructs the decoder to switch to the transparent mode.

3
4 The present invention provides a method for handling data transmissions
5 within a modem that is operating under the V.42*bis* standard.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment of the present invention, a codeword is generated by compressing one or more incoming characters. The incoming characters and codeword are subsequently stored in a buffer. Then, a cost difference between transmitting all the characters previously stored in the buffer and transmitting all the codewords previously stored in the buffer is determined. If the determined cost difference is less than a low limit value, the characters previously stored in the buffer are transmitted. However, if the determined cost difference is greater than a high limit value, the codewords previously stored in the buffer are transmitted. Otherwise, if the determined cost difference falls inclusively between the low limit value and the high limit value, transmission of the data in the buffer is deferred until the cost difference is recalculated on the next codeword.

All features and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention itself, as well as a preferred mode of use, further objects, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 is a high-level logic flow diagram of a method for handling data transmissions within a modem, in accordance with a preferred embodiment of the present invention; and

Figure 2 is a block diagram of a modem in which a preferred embodiment of the present invention can be incorporated.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Generally speaking, the compression ratio achieved by a compression algorithm depends on the data compressibility. If the data in question is not compressible, the compression algorithm not only does not provide compression but may lead to data expansion instead. Hence, when a modem is confronted with a group of highly incompressible data, it is advantageous for the modem to transmit the incoming data without any compression at all. Otherwise, the incoming data should be compressed by a compression algorithm within the modem before transmission in order to achieve a higher transmission efficiency. As such, it is important for an encoder within a modem to correctly decide whether uncompressed or compressed data should be transmitted.

The V.42*bis* standard, which is promulgated by the International Telecommunications Union (ITU) for data transmissions over a telephone network, includes two transmission modes—a transparent mode and a compressed mode. Under the V.42*bis* standard, uncompressed data are being transmitted during the transparent mode, and compressed data are being transmitted during the compressed mode. While the V.42*bis* standard defines most of its implementation requirements, many aspects of the standard are not clearly specified. For example, the V.42*bis* standard does not specify when and how an encoder may decide to switch from a transparent mode to a compressed mode and vice versa. As a result of such ambiguity and under-specification, modems with different implementations of the V.42*bis* standard can have varying levels of compression efficiency. Be that as it may, since the V.42*bis* standard does not constrain when an encoder can switch between transmission modes, there is ample room for improving data transmission efficiency by making intelligent mode selections at the most opportune time.

One prior art implementation of the V.42*bis* standard include two buffers, one with uncompressed data and the other with compressed data. When one of the buffers is filled up, a decision (or guess) is made as to whether it is more efficient to transmit the

1 uncompressed data or the compressed data from their respective buffer in a corresponding
2 transmission mode. But such implementation does not always guarantee optimal
3 transmission efficiency for all data at all times because of the size arbitrariness of the
4 buffer.

5
6 In accordance with a preferred embodiment of the present invention, a mode
7 selection decision is made on a per codeword basis. Specifically, for each codeword, the
8 codeword and corresponding transparent characters are stored in a buffer and then a
9 decision is made as to whether the most efficient transmission mode can be conclusively
10 determined or there is insufficient information to conclude if it would be more efficient to
11 send the buffered codewords in a compressed mode or their corresponding characters in a
12 transparent mode. In the former case, a transmission mode selection is made and all
13 previously buffered data for the chosen transmission mode are sent via the chosen
14 transmission mode. In the latter case, the codeword along with the associated characters
15 are left in the buffer until more information is available.

16
17 A cost model is utilized to assist the above-mentioned decision making
18 process. The cost model includes a cost difference, a low limit value and a high limit
19 value. Preferably, the low limit value is set at 0, and the high limit value is set at the cost
20 (in bits) of switching to the compressed mode and back to the transparent mode. If the cost
21 difference of all buffered codewords is less than the low limit value, the cost model can
22 conclude that it is more beneficial to send buffered characters in the transparent mode. If
23 the cost difference of all buffered codewords is greater than the high limit value, the cost
24 model can conclude that it is more beneficial to send buffered codewords in the compressed
25 mode. If the cost difference falls inclusively between the low and the high limit values
26 (*i.e.*, greater than the low limit value but less than the high limit value or equal to one of
27 the limit values), the cost model cannot conclusively decide which transmission mode to
28 use to send buffered data, and the codewords along with their associated characters will be
29 left in the buffer pending for a subsequent decision.

1 Referring now to the drawings and, in particular, to Figure 1, there is
2 depicted a high-level logic flow diagram of a method for handling data transmissions within
3 a modem, in accordance with a preferred embodiment of the present invention. Starting
4 at block 10, an incoming character is initially received by an encoder within a modem, and
5 the incoming character is compressed to yield a codeword, as shown in block 11. A
6 determination is then made as to whether or not the codeword has been completed, as
7 depicted in block 12. If the codeword has not been completed, the incoming character is
8 sent to a buffer, as depicted in block 18. Otherwise, if the codeword has been completed,
9 a cost difference (*i.e.*, cost.difference) is updated and the codeword is sent to the buffer,
10 as shown in block 13. Cost.C2 is the current number of bits per codeword, as defined in
11 the V.42bis standard.

12
13 Next, a determination is made as to whether or not the cost difference is less
14 than the low limit value (which is preferably set to 0), as depicted in block 14. If the cost
15 difference is less than the low limit value, then the transmission mode is set to the
16 transparent mode, the high limit value (*i.e.*, cost.modechange or the transmission cost of
17 switching to the compressed mode and back) is calculated, the cost difference is set to zero,
18 and the character(s) previously stored in the buffer is transmitted via the transparent mode,
19 as shown in block 15. Otherwise, if the cost difference is greater than the low limit value,
20 then another determination is made as to whether or not the cost difference is greater than
21 the high limit value, as depicted in block 16. If the cost difference is not greater than the
22 high limit value, then the incoming character is sent to the buffer, as depicted in block 18.
23 Otherwise, if the cost difference is greater than the high limit value, then the transmission
24 mode is set to the compressed mode, a new high limit value is calculated, the cost
25 difference is set to the new high limit value, and the codeword(s) previously stored in the
26 buffer is transmitted via the compressed mode, as shown in block 17. After the incoming
27 character has been sent to the buffer, and the cost difference has been incremented (by
28 sixteen if the incoming character is an escape character, or by eight otherwise), as depicted
29 in block 18, the process returns to block 11 to receive a new character.

1 When the cost difference falls inclusively between the low limit value and
2 the high limit value (*i.e.*, the *no* decision path for block 16 in Figure 1), the cost difference
3 is in a "transition region" and the goal is to leave the transition region such that a mode
4 selection decision can be reached as soon as practical. If the cost difference is in the
5 transition region and the modem needs to be flushed, the current transmission mode can be
6 utilized to send the corresponding data previously stored in the buffer because when there
7 is no more data to be sent, saving a few bits is not worth any real effort to come up with
8 a mode selection decision. Of course, a flush operation should only be initiated when there
9 is no more data to send to the far side, and it is used to insure that all previously
10 compressed data is output to the receiving modem. Thus, an alternative method is to
11 always switch to the transparent mode when the modem needs to be flushed. Such strategy
12 allows the most efficient compressor operation because compressed strings are not broken
13 at arbitrary boundaries.

14
15 If the cost difference is in the transition region and the modem does not need
16 to be flushed, the most efficient method is to hold off on deciding which transmission mode
17 to use to send the current data, and see if the immediate subsequent data-to-send changes
18 the cost difference sufficiently to take the cost difference out of the transition region. In
19 order to support the delay on decision making, a relatively large buffer is required. The
20 present invention operates most efficiently if the buffer for storing codewords and their
21 corresponding characters is large enough such that it will never be filled up. However,
22 there are several options that can be taken if the buffer happens to be filled up. A buffer
23 of 1024 words can hold, for example, at least 512 uncompressed bytes. Even if a wrong
24 transmission mode selection is made, the highest cost to get back to the correct transmission
25 mode is approximately 16 bits (could be a little more when switching from the compressed
26 mode to the transparent mode), which means the penalty for making a wrong transmission
27 mode decision would be at most $16/(512 \times 8)$ or approximately 0.4% of the data where the
28 best transmission mode is inconclusive. In practice, runs of inconclusive data do not

1 usually last that long, and the buffer size can be made much smaller without significantly
2 impacting the compression ratio on most types of data.

3
4 When the buffer does fill up, even though there is no clear-cut answer as to
5 whether it is more beneficial to use the compressed mode or the transparent mode to send
6 the corresponding data, at least some data must be sent. The following are options for
7 dealing with a full buffer:

- 8
- 9 (1) Simply use the current transmission mode to send the entire group of the
10 corresponding data in the buffer. If the buffer is substantially large, this option will
11 not be very detrimental to the overall compression ratio.
12
 - 13 (2) Check the current cost difference and use whatever transmission mode that yields
14 the lowest cost at the present time to send the entire group of corresponding data
15 in the buffer.
16
 - 17 (3) Check the current cost difference and select a transmission mode based on the
18 current cost difference and other heuristic information (such as the current
19 transmission mode or transparent/compressed history information) to send the entire
20 group of corresponding data in the buffer. If there is no more data coming into an
21 encoder, this option will perform worse than option (2). But when there are more
22 data coming into the encoder, the overall throughput may be increased.
23
 - 24 (4) Use the current transmission mode to send a portion of the corresponding data in
25 the buffer, recalculate the current cost difference for the remaining data in the
26 buffer, and hold off on sending the remaining data in the buffer until the cost
27 difference is no longer in the transition region or the buffer fills up again.
28

1 In general, option (4) is the most complex, especially if the modem is attempting to
2 maintain a highly accurate cost model. Each character requires 8 or 16 bits to transmit,
3 depending on whether it is an escape code or not, and any particular codeword may require
4 more bits to transmit than the previous one (because of a step up in the codeword size).
5 This means that the additional information must be buffered along with the characters and
6 the codewords, or they must be reconstructed whenever a portion of the corresponding data
7 in the buffer has been transmitted. It also means that the encoder must have a method to
8 determine an intermediate point in the buffer to indicate when to stop sending data. For
9 options (1)–(3), once the encoder decides to transmit the data from the buffer, the encoder
10 can output the entire group of data stored in the buffer before accepting any more data into
11 the buffer.

12
13 The following are some examples for illustrating a preferred embodiment of
14 the present invention. The examples assume a codeword size of 10 bits. Each codeword
15 encodes one or more 8-bit characters (the number of characters represented is a function
16 of the compressibility of the source data and a function of how the V.42bis dictionary
17 works). Also, switching from the transparent mode to the compressed mode (*i.e.*, ESC
18 character followed by ECM character) costs 16 bits, and switching from the compressed
19 to the transparent mode (*i.e.*, ETM codeword) costs 10 bits plus the number of bits it takes
20 to become byte-aligned.

21 22 Example 1

23 Assume the encoder is currently in the transparent mode. The next 10-bit codeword to be
24 sent encodes a single 8-bit character. Result: The eight bits are transmitted immediately
25 because there is no possible benefit in switching to the compressed mode and then sending
26 ten bits instead of sending eight bits in the transparent mode.

Example 2

Assume the encoder is currently in the compressed mode. The next codeword to be sent encodes two 8-bit characters. Result: The 10-bit codeword is transmitted immediately because there is no possible benefit in switching to the transparent mode and then sending sixteen bits instead of ten bits in the compressed mode.

Example 3

Assume the encoder is currently in the transparent mode. The next codeword to be sent encodes six 8-bit characters. Result: The encoder switches to the compressed mode, and then transmits the current codeword in the compressed mode. This is demonstrably the most efficient course of action even if the codewords immediately following the current codeword are better sent in the transparent mode, because it will cost less to switch to the compressed mode (16 bits), send the current codeword (10 bits) switch back (10 bits) and realign to a byte boundary (4 bits) than it will cost to stay in the transparent mode and output the six transparent bytes (40 bits verse 48 bits).

Example 4

Assume the encoder is currently in the transparent mode. The next codeword to be sent encodes four 8-bit characters. Result: The encoder should do nothing until it has more data, because it does not have enough information to know which transmission mode the current data should be sent in. This is because if the encoder were to stay in the transparent mode, it would take 32 bits to send the data, but if the encoder were to switch to the compressed mode to send the data, it would only take 26 bits (16 bits to switch to the compressed mode, and 10 bits to send subsequent data). However, if subsequent data were not compressible, the encoder would have to switch back to the transparent mode to send the subsequent data, which would cost another 14 bits (10 bits to switch back, plus 4 bits to align to a byte boundary) for a net loss of 8 bits.

Example 5

Assume that the encoder has buffered and not yet sent a codeword that encodes four 8-bit characters, as in Example 4, and the next codeword to be sent encodes two 8-bit characters. At this point, it would cost the same amount, 48 bits, to transmit the data in transparent mode (six 8-bit characters) as in compressed mode (16 bits to switch to the compressed mode, 20 bits to send two 10-bit codewords, and 12 bits to switch back to the transparent mode (10 + 2 alignment bits)). At this point, there are three options:

- a) Simply output data in the transparent mode. This will not be optimal if the codeword following these two is better sent in the compressed mode.
- b) Simply output data in the compressed mode. This will send the optimal number of bits in any case, but may incur an extra two mode transitions (if the codewords following these two are best sent in the transparent mode). For various efficiency reasons for both the encoder and decoder, it may be useful to reduce the number of mode transitions.
- c) Delay until more data to yield a clear-cut answer. This will send the optimal number of bits unless the buffer fills up because of such indecisiveness. This may be the best option unless the buffer is very small or unless it is more desirable to reduce buffering as much as possible for latency reasons.

Option (c) is the one used by the preferred embodiment shown in Figure 1. One skilled in the art could choose one of the other options as desired by changing the calculation of the limit values and/or by using " \leq " and " \geq " in preference to "<" and ">" in the comparisons in Figure 1.

Example 6

Assume the same situation as in Example 5, except that the next codeword to be sent encodes three 8-bit characters, instead of two 8-bit characters. Result: Switch to the

1 compressed mode and send the two codewords corresponding to the seven transparent bytes.
2 This is because even if the encoder has to switch back to the transparent mode immediately,
3 sending codewords in the compressed mode costs only 48 bits (verses 56 bits in transparent
4 mode).

5
6 With reference now Figure 2, there is depicted a block diagram of a modem
7 in which a preferred embodiment of the present invention can be incorporated. As shown,
8 modem 20 includes a system-side line-isolation module 21 and a line-side line-isolation
9 module 31. System-side line-isolation module 21 and line-side line-isolation module 31 are
10 coupled to each other via an isolation barrier 30 such as a capacitor. System-side
11 line-isolation module 21 includes an isolation interface 23, a multiplexer 24, an universal
12 asynchronous receiver transmitter (UART) 25, a control interface 26, a digital signal
13 processor (DSP) 27, a clock interface 28, and an audio coder-decoder (CODEC) 29.

14
15 System-side line-isolation module 21 includes a transmit pin TXD, a receive
16 pin RXD, a reset pin RESET*, a clear-to-send pin CTS*, a clock output pin CLKOUT,
17 crystal oscillator pins XTALI and XTALO, and an analog output pin AOUT. System-side
18 line-isolation module 21 also includes four general purpose programmable input/output pins
19 GPIO1, GPIO2, GPIO3, and GPIO4. Each of the above-mentioned four general purpose
20 programmable input/output pins may be set up as analog in, digital in, or digital out pins,
21 depending on user programming of pin functionality. For example, the GPIO1 pin may
22 also function as an end of frame pin EOFR. The GPIO2 pin may provide an analog in pin
23 AIN. The GPIO3 pin may function as an escape pin ESC for controlling command or data
24 modes. The GPIO4 pin may function as an alert pin ALERT for signaling events such as
25 an intrusion event.

26
27 UART 25 converts parallel bytes from DSP 27 into serial bits for
28 transmission to and receipt from an external device through the transmit pin TXD and the
29 receive pin RXD, respectively. In addition to signal processing, DSP 27 also functions as

1 a data pump and provides AT command decoding. Clock interface **28** includes a clock
2 generator and generates all the modem sample rates for supporting the modem standards
3 designed into system-side line-isolation module **21**.

4
5 Line-side line-isolation module **31** includes an isolation interface **32**, a hybrid
6 circuit **33** and a ring detect and off-hook circuit **34**. Hybrid circuit **33** may include a DC
7 termination circuit as well as an analog-to-digital converter (ADC) circuit and a
8 digital-to-analog converter (DAC) circuit. Line-side line-isolation module **31** includes a
9 receive input pin RX, filter pins FILT and FILT2, a reference pin REF, a DC termination
10 pin DCT, voltage regulation pins VREG and VREG2, external resistor pins REXT and
11 REXT2, ring pins RNG1 and RNG2 and transistor connection pins QB, QE and QE2.

12
13 Filter pins FILT and FILT2 set the time constant for the DC termination
14 circuit. The reference pin REF may be connected to an external resistor to provide a high
15 accuracy reference current. The DC termination pin DCT provides DC termination for a
16 telephone phone line and an input for voltage monitors. Voltage regulation pins VREG and
17 VREG2 may be connected to external capacitors to provide a bypass internal power supply.
18 External resistor pins REXT and REXT2 provide real and complex AC termination. Ring
19 pins RNG1 and RNG2 may be connected through capacitors to "tip" and "ring" to provide
20 ring and caller ID signals across isolation barrier **30**. Transistor connection pins QB, QE
21 and QE2 may be connected to external bipolar hook-switch transistors.

22
23 As has been described, the present invention provides a method for
24 determining optimal switching between transmission modes, such as transparent and
25 compressed modes, when transmitting data within a modem that is operating under the
26 V.42*bis* standard.

27
28 While the invention has been particularly shown and described with reference
29 to a preferred embodiment, it will be understood by those skilled in the art that various

1 changes in form and detail may be made therein without departing from the spirit and scope
2 of the invention.